# KX: A flexible system for Keyphrase eXtraction

**Emanuele Pianta**
Fondazione Bruno Kessler
Trento, Italy.
pianta@fbk.eu

**Sara Tonelli**
Fondazione Bruno Kessler
Trento, Italy.
satonelli@fbk.eu

## Abstract

In this paper we present KX, a system for keyphrase extraction developed at FBK-IRST, which exploits basic linguistic annotation combined with simple statistical measures to select a list of weighted keywords from a document. The system is flexible in that it offers to the user the possibility of setting parameters such as frequency thresholds for collocation extraction and indicators for keyphrase relevance, as well as it allows for domain adaptation exploiting a corpus of documents in an unsupervised way. KX is also easily adaptable to new languages in that it requires only a PoS-Tagger to derive lexical patterns. In the SemEval task 5 "Automatic Keyphrase Extraction from Scientific Articles", KX performance achieved satisfactory results both in finding *reader-assigned keywords* and in the *combined keywords* subtask.

## 1 Introduction

Keyphrases are expressions, either single words or phrases, describing the most important concepts of a document. As such, a list of keyphrases provides an approximate but useful characterization of the content of a text and can be used in a number of interesting ways both for human and automatic processing. For example, keyphrases provide a sort of quick summary of a document. This can be exploited not only in automatic *summarization* tasks, but also to enable quick *topic search* over a number of documents indexed according to their keywords, which is more precise and efficient than full-text search. Once the keywords of a document collection are known, they can also be used to calculate *semantic similarity* between documents and to *cluster* the texts according to such similarity (Ricca et al, 2004). Also, keyword extraction can be used as an intermediate step for *automatic sense extraction* (Jones et al, 2002).

For these reasons, the keyphrase extraction task proposed at SemEval 2010 raised much attention among NLP researchers, with 20 groups participating to the competition. In this framework, we presented the KX system, specifically tuned to identify keyphrases in scientific articles. In particular, the challenge comprised two subtasks: the extraction of *reader-assigned* and of *author-assigned* keyphrases in scientific articles from the ACM digital library. The former are assigned to the articles by annotators, who can choose only keyphrases that occur in the document, while author-assigned keyphrases are not necessarily included in the text.

## 2 KX architecture

A previous version of the KX system, named KXPat (Pianta, 2009), was developed to extract keyphrases from patent documents in the PatExpert project (`www.patexpert.org`). The system employed in the SemEval task has additional parameters and has been tailored to identify keyphrases in scientific articles.

With KX, the identification of keyphrases can be accomplished with or without the help of a reference corpus, from which some statistical measures are computed in an unsupervised way. We present here the general KX architecture, including the corpus-based pre-processing, even if in the SemEval task the information extracted from the corpus did not contribute as expected (see Section 3).

KX keyphrase extraction combines linguistic and statistical information, similar to (Frantzi et al., 2000) and is based on 4 steps. The first three steps are carried out at corpus level, whereas the fourth one extracts information specific to each single document to be processed. This means that the first three steps require a corpus *C*, preferably sharing the same domain of the document *d* from which the keyphrases should be extracted. The fourth step, instead, is focused only on the

*Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, pages 170–173,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

document *d*. The steps can be summarized as follows:

- Step 1: Extract from *C* the list *NG-c* of *corpus* n-grams, where an n-gram is any sequence of tokens in the text, for instance "the system", "of the", "specifically built".
- Step 2: Select from the list *NG-c* a sub-list of multiword terms *MW-c*, that is combinations of words expressing a unitary concept, for instance "light beam" or "access control"
- Step 3: For each document in *C*, recognize and mark the multiword terms. Calculate the inverse document frequency (IDF) for all words and multiword terms in the corpus.
- Step 4: Given a document *d* from which a set of relevant keyphrases should be extracted, count all words and multiword terms and rank them.

Step 1 is aimed at building a list of all possible n-grams in *C*. The maximum length of the selected n-grams can be set by the user. For SemEval, beside one-token n-grams, we select 2-, 3- and 4-grams. Since n-grams occurring only a few times are very unlikely to be useful for keyphrase recognition, they are cut off from the extracted list and excluded for further processing. The frequency threshold can be set according to the reference corpus dimensions. For SemEval, we fixed the frequency threshold to 4. In this step, a black-list was also used in order to exclude n-grams containing any of the stopwords in the list. Such stopwords include for example "everything", "exemplary", "preceding", etc.

In Step 2, we select as multiword terms those n-grams that match certain lexical patterns. To this purpose, we first analyze all n-grams with the MorphoPro morphological analyzer of the TextPro toolsuite (Pianta et al., 2006). Then, we filter out the n-grams whose analysis does not correspond to a predefined set of lexical patterns. For example, one of the patterns admitted for 4-grams is the following: [N]~[O]~[ASPGLU]~[NU]. This means that a 4-gram is a candidate multiword term if it is composed by a Noun, followed by "of" or "for" (defined as O), followed by either an Adjective, Singular noun, Past participle, Gerund, punctuation (L) or Unknown word, followed by either a Noun or Unknown word. This is matched for example by the 4-gram "subset [S] of [O] parent [S] peers [N]". Both the lexical categories (e.g. S for singular noun) and the admissible lexical patterns can be defined by the user.

In Step 3, multiword terms are recognized by combining local (document) and global (corpus) evidence. To this purpose, we do not exploit association measures such as Log-Likelihood, or Mutual Information, but a simple frequency based criterion. Two thresholds are defined: *MinCorpus*, which corresponds to the minimum number of occurrences of an n-gram in a *reference corpus*, and *MinDoc*, i.e. the minimum number of occurrences in the *current document*. KX marks an n-gram in a document as a multiword term if it occurs at least *MinCorpus* times in the corpus or at least *MinDoc* times in the document. The two parameters depend on the size of the corpus and the document respectively. In SemEval, we found that the best thresholds are *MinDoc*=4 and *MinCorpus*=8. A similar, frequency-based, strategy is used to solve ambiguities in how sequences of contiguous multiwords should be segmented. For instance, given the sequence "combined storage capability of sensors" we need to decide whether we recognize "combined storage capability" or "storage capability of sensors". To this purpose, we calculate the strength of each alternative collocation as *docFrequency * corpusFrequency*, and then choose the stronger one. To calculate IDF for each word and multiword term, we use the usual formula: *log( TotDocs / DocsContaningTerm )*.

In step 4, we take into account a new document *d*, possibly not included in *C*, from which the keyphrases should be extracted. First we recognize and mark multiword terms, through the same algorithm used in Step 3. Note that KX can recognize multiwords also in isolated documents, independently of any reference corpus, by activating only the *MinDoc* parameter (see above). Then, we count the frequency of words and multiword terms in *d*, obtaining a first list of keyphrases, ranked according to frequency. Thus, *frequency* is the *baseline* ranking parameter, based on the assumption that important concepts are mentioned more frequently than less important ones.

After the creation of a frequency-based list of keyphrases, various techniques are used to re-rank it according to relevance. In order to find the best ranking mechanism for the type of keyphrases we want to extract, different parameters can be set:

- *Inverse document frequency* (*IDF*): this parameter takes into account the fact that a concept that is mentioned in all documents is less relevant to our task than a concept occurring in few documents

- *Keyphrase length*: number of tokens in a keyphrase. Concepts expressed by longer phrases are expected to be more specific, and thus more relevant. When this parameter is activated, frequency is multiplied by the keyphrase length.
- *Position of first occurrence*: important concepts are expected to be mentioned before less relevant ones. If the parameter is activated, the frequency score will be multiplied by the *PosFact* factor computed as $(DistFromEnd\ /\ MaxIndex)^{pwr2}$, where *MaxIndex* is the length of the current document and *DistFromEnd* is *MaxIndex* minus the position of the first keyphrase occurrence in the text.
- *Shorter concept subsumption*: In the keyphrase list, two concepts can occur, such that one is a specification of the other. Concept subsumption and boosting are used to merge or re-rank such couples of concepts. If a keyphrase is (stringwise) included in a longer keyphrase with a *higher frequency*, the frequency of the shorter keyphrase is transferred to the count of the longer one. E.g. "grid service discovery"=6 and "grid service"=4 are re-ranked as "grid service discovery"=10 and "grid service"=0
- *Longer concept boosting:* If a keyphrase is included in a longer one with a *lower frequency*, the average score between the two keyphrase frequency is computed. Such score is assigned to the less frequent keyphrase and subtracted from the frequency score of the higher ranked one. For example, if "grid service discovery"=4 and "grid service"=6, the average frequency is 5, so that "grid service discovery"=5 and "grid service" = 6–5=1. This parameter can be activated alone or together with another one that modifies the criterion for computing the boosting. With this second option, the longer keyphrase is assigned the frequency of the shorter one. For example, if "grid service discovery"=4 and "grid service"=6, the boosting gives "grid service discovery"=6 and "grid service"=6.

After the list of ranked keyphrases is extracted for each document, it is finally *post-processed* in two steps. The post-processing phase has been added specifically for SemEval, because keyphrases do not usually need to be stemmed and acronym expansion is relevant only for the specific genre of scientific articles. For this reason, the two processes are not part of the official system architecture.

First, acronyms are replaced by the extended form, which is automatically extracted from the current document. The algorithm for acronym detection scans for parenthetical expressions in the text and checks if a preceding text span can be considered a suitable correspondence (Nguyen and Kan, 2007). The algorithm should detect cases in which the acronym appears after or before the extended form, like in "Immediate Predecessors Tracking (IPT)" and "IPT (Immediate Predecessors Tracking)". If the acronym and the extended form appear both in the keyphrase list, only the extended form is kept and the acronym frequency is added.

The second step is stemming with the (Porter Stemmer). Then, we check if the list of stemmed keyphrases contains duplicate entries. If yes, we sum the frequencies of the double keyphrases and remove one of the two from the list.

## 3 Experimental Setup

In the SemEval task, 144 training files were made available before the test data release. We split them into a training/development set of 100 documents and a test set of 44 documents, in order to find the best parameter combination. Keyphrase assignment is a subjective task and criteria for keyphrase identification depend on the domain and on the goal for which the keyphrases are needed. For example in scientific articles longer keyphrases are often more informative than shorter ones, so the parameters for boosting longer concepts are particularly relevant.

We first tested all parameters in isolation to compute the improvement over the frequency-based baseline. Results are reported in Table 1. F1 is computed as the harmonic mean of precision and recall over the 15 top-ranked keyphrases after stemming. We report the *combined F1*, as computed by the task scorer in order to combine *reader-assigned* and *author-assigned* keyword sets.

| Parameter | F1 (combined) |
|---|---|
| Baseline(MinDoc = 2) | 13.63 |
| Baseline(MinDoc = 4) | 14.84 |
| +CorpusColloc(small) | 13.48 |
| +CorpusColloc(big) | 13.33 |
| +IDF | 17.98 |

| | |
|---|---|
| +KeyphraseLength | 16.78 |
| +FirstPosition | 16.18 |
| +ShortConcSubsumption | 16.03 |
| +LongConcBoost(version1) | 14.38 |
| +LongConcBoost(version2) | 13.93 |
| MinDoc = 4, +FirstPosition, +IDF, +KeyphraseLength, +ShortConcSubsumption, +LongConcBoost(version1) | **25.62** |

Table 1: Parameter performance over development set

The parameter scoring the highest improvement over the baseline is IDF. Also the parameters boosting longer keyphrases and those that occur at the beginning of the text are effective. Note that the *LongConcBoost* parameter achieves better results in the first version, which has a higher impact on the re-ranking. Surprisingly, using a domain corpus to extract information about multiword terms, as described in Section 2, steps 1 - 3, does not achieve any improvement. This means that KX can better recognize keyphrases in single documents without any corpus reference. Besides, the best setting for *MinDoc*, the minimum number of multiword occurrences in the current document (see Section 2) is 4. We tested the *CorpusColl*c parameter using two different reference corpora: one contained the 100 articles of the training set (*CorpusColloc small*), while the other (*CorpusColloc big*) included both the 100 training articles and the 200 scientific publications of the NUS Keyphrase Corpus (Nguyen and Kan, 2007). The performance is worse using the larger corpus than the smaller one, and in both cases it is below the baseline obtained without any reference corpus.

In the bottom row of Table 1, the best parameter combination is reported with the score obtained over the development set. The improvement over the baseline reaches 11.99 F1.

## 4 Evaluation

In the SemEval task, the system was run on the test set (100 articles) with the best performing parameter combination described in the previous section. The results obtained over the 15 top-ranked keyphrases are reported in Table 2.

| Keyphrase type | P | R | F1 |
|---|---|---|---|
| Reader-assigned | 20.33 | 25.33 | **22.56** |
| Combined | 23.60 | 24.15 | **23.87** |

Table 2: System performance over test set

In the competition, the F1 score over reader-assigned keyphrases was ranked 3rd out of 20 participants, while the combined measure achieved the 7th best result out of 20.

## 5 Conclusions

In this work we have described KX, a flexible system for keyphrase extraction, which achieved promising results in the SemEval task 5. The good KX performance is due to its adaptable architecture, based on a set of parameters that can be tailored to the document type, the preferred keyphrase length, etc. The system can also exploit multiword lists (with frequency) extracted from a reference corpus, even if this feature did not improve KX performance in this specific task. However, this proved to be relevant when applied to keyphrase extraction in the patent domain, using a large domain-specific corpus of 10.000 very long documents (Pianta, 2009).

A limitation of KX in the task was that it extracts only keyphrases already present in a given document, while the *author-assigned* subtask in the SemEval competition included also keyphrases that do not occur in the text. Another improvement, which is now being implemented, is the extraction of the best parameter combination using machine-learning techniques.

## References

Jones, S., Lundy, S. and Paynter, G. W. 2002. Interactive Document Summarization Using Automatically Extracted Keyphrases. In *Proc. of the 35th Hawaii International Conference on System Sciences*.

Frantzi, K., Ananiadou, S. and Mima, H. 2000. Automatic recognition of multi-word terms: the C-value/NC-value method. *Journal on Digital Libraries.* 3 (2), pp.115-130.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase Extraction in Scientific Documents. In D.H.-L. Goh et al. (eds.): *ICADL 2007*, LNCS 4822, pp. 317-326.

Pianta, E., Girardi, C and Zanoli, R. 2006. The TextPro tool suite. In *Proc. of LREC*.

Pianta, E. 2009. *Content Distillation from Patent Material*, FBK Technical Report.

Ricca, F., Tonella, P., Girardi, C and Pianta, E. 2004. An empirical study on Keyword-based Web Site Clustering. In *Proceedings of the 12th IWPC*.

PorterStemmer: `http://tartarus.org/~martin/PorterStemmer/perl.txt`.